Lab 3

Exercise 1

Write a query to return the product ID of each product, together with the product name formatted as upper case and a column named ApproxWeight with the weight of each product rounded to the nearest whole unit. Make sure to use the aliases provided, and default column names elsewhere.

Complete the following query:

```
-- select ProductID and use the appropriate functions with the appropriate columns
SELECT ___, ___(___) AS ProductName, ___(___, 0) AS ApproxWeight
FROM SalesLT.Product;
```

Exercise 2

Extend your query to include columns named SellStartYear and SellStartMonth containing the year and month in which AdventureWorks started selling each product. The month should be displayed as the month name (e.g. 'January'). Make sure to use the aliases provided, and default column names elsewhere.

Complete the following query:

```
SELECT ProductID, UPPER(Name) AS ProductName, ROUND(Weight, 0) AS ApproxWeight,
       -- get the year of the SellStartDate
       ___(___) as SellStartYear,
       -- get the month datepart of the SellStartDate
       ___(m, ___) as SellStartMonth
FROM SalesLT.Product;
```

Exercise 3

The sales manager would like a list of customers ranked by sales. Write a query that returns a list of company names with a ranking of their place in a list of highest TotalDue values from the SalesOrderHeader table. Make sure to use the aliases provided, and default column names elsewhere.

Complete the following query:

```
-- select CompanyName and TotalDue columns
SELECT ___, ___ AS Revenue,
       -- get ranking and order by appropriate column
       ___ OVER (ORDER BY ___ DESC) AS RankByRevenue
FROM SalesLT.SalesOrderHeader AS SOH
-- use appropriate join on appropriate table
___ SalesLT.Customer AS C
ON SOH.CustomerID = C.CustomerID;
```

Exercise 4

AdventureWorks products each have a standard cost that indicates the cost of manufacturing the product, and a list price that indicates the recommended selling price for the product. This data is stored in the SalesLT.Product table. Whenever a product is ordered, the actual unit price at which it was sold is also recorded in the SalesLT.SalesOrderDetail table. Use subqueries to compare the cost and list prices for each product with the unit prices charged in each sale.

Retrieve the product ID, name, and list price for each product where the list price is higher than the average unit price for all products that have been sold.

Complete the following query:

```
-- select the ProductID, Name, and ListPrice columns
SELECT ___, ___, ___
FROM SalesLT.Product
-- filter based on ListPrice
WHERE ___ >
-- get the average UnitPrice
(SELECT ___ FROM SalesLT.SalesOrderDetail)
ORDER BY ProductID;
```

Exercise 5

AdventureWorks is interested in finding out which products are being sold at a loss. Retrieve the product ID, name, and list price for each product where the list price is 100 or more, and the product has been sold for (strictly) less than 100.
Remember, the ProductID in your subquery will be from the SalesLT.SalesOrderDetail table.

Complete the following query:

```
SELECT ProductID, Name, ListPrice
FROM SalesLT.Product
WHERE ProductID IN
  -- select ProductID from the appropriate table
  (SELECT ___ FROM SalesLT.___
   WHERE UnitPrice < 100)
AND ListPrice >= 100
ORDER BY ProductID;
```